1. Introduction

We will discuss the wave equation. The problem is, there are so many of them. This is a wave equation:

\[ u_{xx} = u_{tt}. \]

So is this:

\[
\begin{bmatrix}
M \frac{\partial}{\partial t} \begin{bmatrix} p \\ u \end{bmatrix} + D_1 \frac{\partial}{\partial x} \begin{bmatrix} p \\ u \end{bmatrix} + D_2 \frac{\partial}{\partial y} \begin{bmatrix} p \\ u \end{bmatrix} + D_3 \frac{\partial}{\partial z} \begin{bmatrix} p \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix},
\end{bmatrix}
\]

as is this:

\[ \rho \frac{\partial^2 u}{\partial t^2} = \nabla \cdot (\mu \nabla u) + \nabla ((\lambda + \mu) \nabla \cdot u) + F. \]

And that’s just a few of them.

In these notes, we will examine a few instances of the wave equation, beginning with a one-way wave equation in one spatial dimension, and progressing eventually to the elastic wave equation in 2 and 3 spatial dimensions. We will pay particular attention to where the physical coefficients like density and elasticity appear in the equations, which is important for inhomogeneous media. Along the way, we will discuss exact and numerical methods for solution, and wrap up with some numerical examples of wave field propagators.

2. One-way wave equation, 1D

The traffic flow model is a simple one-dimensional system that demonstrates wave-like solutions. Consider a straight stretch of highway where cars are flowing in one direction, at various speeds \( c(x) \) that depend on the position \( x \) along the road (for instance, there may be different speed limits set on different portions of the road). From our own experience, we know that cars will cluster up and spread out in response to the change in speed limits. This clustering effect will move as waves along the highway.

To be more precise, imagine that measuring devices are placed along the road that record the rate at which cars are passing various spots in the road. This data is recorded as a function \( u(x, t) \), indicating the number of cars per second that are passing at point \( x \) along...
the road, which may change with time \( t \) as the traffic progresses. The ratio \( u(x, t)/c(x) \) measures the density of the cars at point \( x \) (say in cars/meter), so the integral

\[
\int_a^b \frac{u(x, t)}{c(x)} \, dx
\]

represents the total number of cars in the interval \([a, b]\) along the highway. The time derivative of this integral tells us the net change in number of cars in the interval, which can only occur because cars enter at point \( x = a \) and exit at point \( x = b \). This is given by the difference of the rate of flow of cars at those two points, so we have

\[
\frac{\partial}{\partial t} \int_a^b \frac{u(x, t)}{c(x)} \, dx = u(a, t) - u(b, t).
\]

The RHS can be expressed as an integral of the \( x \)-derivative, so

\[
\frac{\partial}{\partial t} \int_a^b \frac{u(x, t)}{c(x)} \, dx = -\int_a^b \frac{\partial u(x, t)}{\partial x} \, dx.
\]

Combining the two integrals and pulling the time derivative inside, we have that for all points \( a, b \) along the highway,

\[
\int_a^b \left[ \frac{1}{c(x)} \frac{\partial u(x, t)}{\partial t} + \frac{\partial u(x, t)}{\partial x} \right] \, dx = 0.
\]

Assuming continuing of the derivatives, this can only happen if the integral is always zero, in which case we have obtained the first order partial differential equation

\[
\frac{\partial u}{\partial t} + c(x) \frac{\partial u}{\partial x} = 0,
\]

where \( t \) is time, \( x \) is displacement in space, \( u(x, t) \) is the rate of traffic flow at any position and time, and \( c(x) \) is a velocity of traffic flow.

For simple case of constant velocity \( c(x) \equiv c \), it is easy to check that the general solution is given in the form

\[
u(x, t) = f(x - ct),
\]

where \( f \) is any function, say once-differentiable so that the DE makes sense. But if you are happy to work with weak solutions, \( f \) could be any measurable function.

So what we see is that the function \( f(x) \) represents an initial waveform at time \( t = 0 \) and then the waveform just moves uniformly to the right, at a velocity \( dx/dt = c \). In particular, if \( f \) has any sharp corners or discontinuities, they just propagate along with velocity \( a \) and they do NOT disappear nor do they get smoothed out.

We can also solve for the case of non-constant velocity \( c = c(x) \), which is a bit more instructive as we can see more details of how the wave responds to inhomogeneities in the medium of propagation. Supposing \( c(x) \) is non-zero (and let’s say always positive) and piecewise continuous, we define the slowness \( s \) as the inverse of velocity, with

\[
s(x) = \frac{1}{c(x)},
\]
with antiderivative given as $S(x)$,

$$S(x) = \int_0^x s(x') \, dx'. \quad (8)$$

With our assumptions on $c(x)$, this function $S(x)$ is continuous and piecewise continuously differentiable, and the general solution to our one-way wave equation is found to be

$$u(x,t) = f(t - S(x)), \quad (9)$$

since $\frac{\partial u}{\partial t} = f'(t - S(x))$, $c(x)\frac{\partial u}{\partial x} = -c(x)S'(x)f'(t - S(x)) = -\frac{\partial u}{\partial t}$, as desired.

Again, this solution makes sense for any function $f$ which is once-differentiable, but it also can be extends as a weak solution, for $f$ merely measurable.

What’s more, we see that the characteristic curves $t - S(x) = 'constant' \text{ completely determine the solution } u(x,t) \text{ since the function } u \text{ is necessarily constant along the characteristic curves. Thus, if we make a plot of these characteristic curves in the } x-t \text{ plane, we can visualize the propagation of the wave as follows. It moves to the right (} c(x) > 0 \text{ was assumed) and the speed at which it moves is given by the reciprocal of the slope } S'(x) \text{ of the characteristic curves. We might well call these the “slowness curves.”}

Referring to Figure 1, we see six slowness curves (in black) corresponding to times $t = 0...5$. An initial waveform is displayed (in red) along the horizontal axis $t = 0$. As time passes, the waveform moves to the left, and is displayed at times $t = 0, 5, 10, 15$.

We observe that the initial waveform $u(x,0) = f(-S(x))$ gets deformed as it moves, compressing where the slope of the characteristic curve is steep (slowness is large, velocity is small) and stretching out when that slope is not steep (slowness is small, velocity is large). In particular, while the width of the waveform does vary (and varies inversely with velocity), its amplitude does not change. The waveform shape does not preserve symmetry, as different portions of the wave get compressed more or less, depending on the relative $u(x)$ value at those spots. It is also interesting to note that as the slowness curves flatten out again, the initial shape of the waveform is recovered.

These solutions are exact. Which means that they would be useful for testing the accuracy of any numerical PDE code we might develop or choose to evaluate. Again note we see no dispersion (i.e. no wiggles are introduced) and no dissipation (the waveform does not shrink from a loss of energy). These are typical numerical effects that we might expect to see in numerical PDE algorithms.

Also notice the sharp corners at $x = -1.5$, $x = 0$ in the initial waveform stay nicely preserved under the wave propagation.

3. Where does the parameter velocity go?

We note in the PDE of the last section, the parameter $c(x)$ sat outside the derivative. If we had set up the same traffic problem with $u(x,t)$ representing density of the cars, rather than rate, the governing PDE would look like this:

$$\frac{\partial}{\partial t}(u) + \frac{\partial}{\partial x}(c(x)u) = 0, \quad (10)$$
which has its general solution given as

\begin{equation}
  u(x, t) = \frac{f(t - S(x))}{c(x)}.
\end{equation}

The plot in Figure 2 shows the initial density waveform propagating to the right. Observe that not only does the shape of the waveform compress as it propagates, it also changes amplitude. Physically, this just says that as the cars slow down on the highway, the local density increases (think traffic jam). Again, an interesting result which will be relevant in other physical wave equations we examine.

4. Finite difference solutions to the 1D equation

A reasonable finite difference approximation to the PDE

\begin{equation}
  \frac{\partial u}{\partial t} + c(x) \frac{\partial u}{\partial x} = 0
\end{equation}

has the form

\begin{equation}
  \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} + c(x) \frac{u(x + \Delta x, t) - u(x, t)}{\Delta x} = 0.
\end{equation}
Thus if we know the value of \( u \) at some fixed time \( t \), we can solve for the solution \( u(x, t+\Delta t) \) at time \( t + \Delta t \) by the formula

\[
(14) \quad u(x, t + \Delta t) = u(x, t) - c(x) \frac{\Delta t}{\Delta x} (u(x + \Delta x, t) - u(x, t)).
\]

A numerical method is obtained by slicing up the \( x \) and \( t \) intervals over the range of interest into a uniform grid, and using the above formula to iterate over all points in the grid, obtaining an approximation to that range. The Courant-Friedrichs-Lewy (CFL) condition tells us we cannot expect a stable, convergence formula unless the coefficient

\[
(15) \quad R = -c(x) \frac{\Delta t}{\Delta x}
\]

is smaller than one in absolute value.

A quick test of a numerical solution to the traffic flow problem shows the result for time steps \( \Delta t = 0.01 \) and \( 0.001 \), in Figure 3. While the results look reasonable, we see that for the larger step size, the waveform is both flattened and stretched, the result of numerical dissipation and dispersion. For the finer step size, the result are better – but still we see
the corners of the initial waveform are getting smoothed out – the result of numerical approximation.

![Graph showing traffic flow problem](image)

**Figure 3.** Numerical solution for the traffic flow problem, time steps of 0.01 and 0.001.

Somewhat surprisingly, a slightly different finite difference approximation leads to an unstable numerical method.\(^1\) The only change is that derivative in \(x\) is approximated by a difference quotient at points \(x\) and \(x - \Delta x\). It turns out the instability arises because the wave is propagating to the right (velocity \(c(x)\) is positive); if the wave was propagating to the left, then the first method above would have been unstable.

For this reason, with two-way wave propagation it seems we should use a symmetric difference operator, to ensure we can have stability for both directions of flow. Again, we approximate the PDE in finite difference form, but take the \(x\)-derivate in a symmetric approach.

---

\(^1\)If you haven’t seen an unstable method, try this one out. The method will produce huge errors at the limits of your computer arithmetic in only a few iterations.
WAVE EQUATION, PROPAGATORS, AND NUMERICS

form, giving

\[ \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} + c(x) \frac{u(x + \Delta x, t) - u(x - \Delta x, t)}{2\Delta x} = 0. \]

This gives the iteration for finding \( u(x, t + \Delta t) \) as

\[ u(x, t + \Delta t) = -c(x) \frac{\Delta t}{2\Delta x} (u(x + \Delta x, t) - u(x - \Delta x, t)). \]

Unfortunately, this method is always unstable — no matter what choice of \( \Delta t, \Delta x \) is made.

This method can be stabilized by adding a second order derivative into the iteration. This is called the Lax-Wendroff scheme. Without getting into the theoretical details, the iteration is given by the formula

\[ u(x, t + \Delta t) = -c(x) \frac{\Delta t}{2\Delta x} (u(x + \Delta x, t) - u(x - \Delta x, t)) + c(x)^2 \frac{\Delta t^2}{2\Delta x^2} (u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)). \]

A numerical result for this scheme is shown in Figure 4. Again, we observe a reasonable solution (it is not unstable) but still has the waveform flattened and stretched due to numerical dissipation and dispersion. We will point out, however, that the iteration scheme is symmetric in \( x \), so it would work for both left-going and right-going waves.

5. **More finite difference methods**

We saw in the previous section that the first order PDE \( \partial u / \partial t + c(x) \partial u / \partial x = 0 \) could be solved numerically by estimating the derivatives using finite differences. There are many possible choices for these estimated derivatives: a forward difference in time \( (u(\cdot, t + \Delta t) - u(\cdot, t)) / \Delta t \), a backward difference in time \( (u(\cdot, t) - u(\cdot, t - \Delta t)) / \Delta t \), centred differences in time, and similarly for the \( x \) differences. Every combination gives a potential numerical method — some are stable, some are not. More interesting is the fact that some give explicit methods (the values of solution are computed directly from previously computed values), while other give implicit methods (the values of the solution arise as the solution to a linear system of equations). Implicit methods are computationally more challenging; however this is often pays off in more accurate solutions with fewer grid points in the computation.

The text by Lapidus and Pinder gives an excellent summary of the various finite difference choices and which lead to stable or unstable methods, implicit and explicit solutions.

6. **Other numerical methods**

Finite element and Galerkin methods are commonly used to find numerical solutions to systems of PDEs. For hyperbolic equations such as the wave equation, the standard approaches to these methods do not perform well: oscillations appear in the numerical solutions that should not be there; fixes need to be introduced to damp instabilities, etc.
Lapidus and Pinder state “At this time [1999], the finite element and collocations schemes do not appear to have advanced to the stage where they present a serious threat to the more established finite difference approaches.” This is an area for further study. (See for instance the thesis by McDonald, where discontinuous Galerkin methods are applied to the wave equation.)

### 7. First Order Acoustic Wave Equation

We are well aware that sound travels through a fluid like air or water as waves. A wave equation for sound can be obtained directly by linearization of the Navier-Stokes equation for fluid motion. However, we can leap right in with a physical description that leads directly to a system of first order PDEs describing the acoustic wave motion.

Consider a fluid (in 1D such as in a pipe, 2D as in a layer of fluid, or fully 3D) for which we can measure velocity \( \mathbf{u} = \mathbf{u}(x,t) \) and pressure \( p = p(x,t) \) at any point \( x \) and time \( t \) in the fluid. Conservation of mass tells us the pressure will change in response to any divergence in the velocity field (i.e. net flow out causes the pressure to drop, net flow in
causes the pressure to increase) leading to the equation

\begin{equation}
\frac{\partial p}{\partial t} + K \nabla \cdot \mathbf{u} = 0.
\end{equation}

Here, the coefficient \( K \) is the bulk modulus of the fluid, measuring how strong the fluid pressure changes in response to the divergence of flow. Newton’s second laws tells us the rate of change of momentum is due to forces on the fluid, of which there are two – any gradient in the pressure field, and any external force \( \mathbf{F} \). Thus we have

\begin{equation}
\frac{\partial \mathbf{u}}{\partial t} + \nabla p = \mathbf{F},
\end{equation}

where \( \rho \) is the density of the fluid, \( \rho \frac{\partial \mathbf{u}}{\partial t} \) is the momentum, and the other two terms are forces.

It is important to note that the parameters \( K = K(\mathbf{x}), \rho = \rho(\mathbf{x}) \) are not constant in an inhomogeneous medium and so we have to be careful where they are place with respect to differentiation in the PDEs.

In summary we have a first order system of \( n+1 \) PDEs describing the wave propagation, given as

\begin{equation}
\frac{1}{K} \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} = 0,
\end{equation}

\begin{equation}
\rho \frac{\partial \mathbf{u}}{\partial t} + \nabla p = \mathbf{F}.
\end{equation}

In the 3D case, this can be written out fully as a matrix system of PDEs with

\begin{equation}
\mathbf{M} \frac{\partial}{\partial t} \begin{bmatrix} p \\ \mathbf{u} \end{bmatrix} + \mathbf{D}_1 \frac{\partial}{\partial x} \begin{bmatrix} p \\ \mathbf{u} \end{bmatrix} + \mathbf{D}_2 \frac{\partial}{\partial y} \begin{bmatrix} p \\ \mathbf{u} \end{bmatrix} + \mathbf{D}_3 \frac{\partial}{\partial z} \begin{bmatrix} p \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{F} \end{bmatrix},
\end{equation}

where \( \mathbf{M} \) is a diagonal matrix with entries \((1/K, \rho I)\) and \( \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \) are symmetric matrices with two non-zero entries each.\(^2\) This is a first order hyperbolic system, for which there are robust numerical methods to solve.

8. Second-order acoustic wave equation

To get the familiar second order PDE for the wave equation, we take Eqn (22) and differentiate with respect to time, to obtain

\begin{equation}
\frac{1}{K} \frac{\partial^2 p}{\partial t^2} + \nabla \cdot \frac{\partial \mathbf{u}}{\partial t} = 0,
\end{equation}

and replacing the \( \frac{\partial \mathbf{u}}{\partial t} \) factor using Eqn (23) we obtain

\begin{equation}
\frac{1}{K} \frac{\partial^2 p}{\partial t^2} - \nabla \cdot \left( \frac{1}{\rho} \nabla p \right) = -\frac{\nabla \cdot \mathbf{F}}{\rho}.
\end{equation}

Notice the density parameter \( \rho = \rho(\mathbf{x}) \) appears inside the divergence operator; as we saw with one-way wave equation, it makes a difference for the physics, and for the solution.

\(^{(2)}\)figure them out!
In the special case where density is constant, it can be moved outside the differential and obtain the more familiar form

\[ \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} - \nabla^2 p = -\nabla \cdot F, \]

where the parameter \( c(x) = \sqrt{K(x)/\rho} \) is the local speed of sound.

It is worth noting that once we solve for pressure \( p \), we can recover the velocity field \( u \) simply by integrating Eqn (23), so

\[ u(x,t) = \int_0^t \frac{F(x,s) - \nabla p(x,s)}{\rho(x)} \, ds. \]

Thus the second order equation contains all the information as the system in the previous section.

Alternatively, we differentiate Eqn (23) with respect to time, and substitute \( \partial \rho/\partial t \) from Eqn (22) to obtain

\[ \rho \frac{\partial^2 u}{\partial t^2} - \nabla (K \nabla \cdot u) = \frac{\partial F}{\partial t}. \]

It is interesting to compare this to Eqn (25), where the parameters \( K, \rho \) have changed places, and the forcing term is quite different.

9. A NON-REFLECTING TWO-WAY WAVE EQUATION, IN 1D

Motivated by our one-way equations which are solved explicitly using slowness curves, it would be interesting to find special cases for the two-way equation that has solutions of the form

\[ p(x,t) = f(t \pm S(x)), \]

for arbitrary functions \( f \) and some “slowness” like function \( S(x) \).

In the 1D case, this is easy to achieve. In the special case where the product of density and bulk modulus is constant (say, \( \rho(x)K(x) = \lambda^2 \)), define a function \( S(x) \) as the antiderivative of density

\[ S(x) = \int_0^x \rho(x') \, dx' \]

and let pressure \( p(x,t) = f(\lambda t \pm S(x)) \). Then its second time derivative, divided by bulk modulus, is

\[ \frac{1}{K} \frac{\partial^2 p}{\partial t^2} = \frac{\lambda^2}{K} f''(\lambda t \pm S(x)) = \rho f'', \]
while the $x$-partials give

\begin{align}
\frac{\partial}{\partial x} \left( \frac{1}{\rho} \frac{\partial}{\partial x} p(x,t) \right) &= \frac{\partial}{\partial x} \left( \frac{1}{\rho} \frac{\partial}{\partial x} f(\lambda t \pm S(x)) \right) \\
&= \pm \frac{\partial}{\partial x} \left( \frac{1}{\rho} f'(\lambda t \pm S(x)) \cdot S'(x) \right) \\
&= \pm \frac{\partial}{\partial x} f'(\lambda t \pm S(x)) \\
&= f''(\lambda t \pm S(x)) \cdot S'(x) = \rho f''.
\end{align}

Thus the quantities in Eqns (31) and (32) are equal and so this pressure satisfies the wave equation (27).

In this special case, a right-traveling wave $f(\lambda t - S(x))$ generates no reflections going back to the left, while a left-traveling wave $f(\lambda t + S(x))$ generates no reflections going to the right. This should not be surprising since we understand that seismic imaging typically only “sees” changes in impedance, and the case $\rho(x)K(x)$ constant corresponds to constant impedance.

The wave also undergoes no amplitude changes, and the shape only changes as $S(x)$ changes. There is no dissipation nor dispersion.

This special case is also useful for testing our numerical algorithms as we have exact solutions. As with the one-way wave equation, note that these solutions preserve corners and discontinuities in the wave, which propagate with speed $c = \lambda/S'(x) = \sqrt{K(x)/\rho(x)}$.

10. Non-reflecting two-way wave equation, in 2D

In the previous section, we saw the 1D wave equation

\begin{align}
c(x) \frac{\partial}{\partial x} \left( c(x) \frac{\partial p}{\partial x} \right) &= \frac{\partial^2 p}{\partial t^2}
\end{align}

has left and right going solutions that do not generate reflections. Baysal et al suggest generalizing this PDE to 2D in the form

\begin{align}
c(x,y) \frac{\partial}{\partial x} \left( c(x,y) \frac{\partial p}{\partial x} \right) + c(x,y) \frac{\partial}{\partial y} \left( c(x,y) \frac{\partial p}{\partial y} \right) &= \frac{\partial^2 p}{\partial t^2}.
\end{align}

They show through numerical experiments that normal incident waves do not generate reflections (due to the constant impedance across a boundary), and off-normal incidence waves have a reduced reflection amplitude.

11. Elastic wave equations

A solid body such as a piece of rubber, a block of metal, or a piece of the earth, will transmit vibrational waves. A basic linear model postulates that the stress in the body is a linear function of gradient of the displacement vector $u = u(x,t)$. The stress tensor is thus given by the formula

\begin{align}
\tau = L(\nabla u),
\end{align}

where $L$ is a linear operator.
where \( L \) is a linear function from the 3x3 matrix \( \nabla \mathbf{u} \) to the 3x3 matrix \( \mathbf{T} \). This linear function can be expressed in coordinate form, so

\[
\tau_{ij} = \sum_{k,l} c_{ijkl} \frac{\partial \mathbf{u}_{l}}{\partial x_k},
\]

where the coefficients \( c_{ijkl} \) form the elasticity tensor. (Think of it as a complicated version of Hooke’s law.)

The stress tensor \( \tau \) specifies the traction force per unit area \( \tau \cdot \hat{n} \) on a surface with unit normal \( \hat{n} \). Thus the force on a section of the solid is given by the surface integral

\[
\int \tau \cdot \hat{n} \, dS = \int \nabla \cdot \tau \, dV
\]

which we express as the volume integral on the right hand side by Gauss’ theorem.

Newton’s law says the rate of change of moment is equal to the applied force, for which we obtain

\[
\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \tau + \mathbf{F},
\]

where \( \nabla \cdot \tau \) is the internal force due to the stress, and \( \mathbf{F} \) is any external force.\(^3\)

Combining Eqns (38) and (40) gives the elastic wave equation, in component form, as

\[
\rho \frac{\partial^2 \mathbf{u}_i}{\partial t^2} = \sum_{j,k,l} \frac{\partial}{\partial x_j} \left( c_{ijkl} \frac{\partial \mathbf{u}_l}{\partial x_k} \right) + \mathbf{F}_i.
\]

It is worth noting the general similarity to Eqn (29), in particular the placement of the density parameter \( \rho \) and the elasticity parameters \( K \) and \( c_{ijkl} \).

It would seem that the 81 parameters \( c_{ijkl} \) potentially represent a very rich choice of variables. However, physical symmetry conditions force relations that leave only 21 independent parameters. When the material is isotropic, there are only two independent parameter \( \lambda, \mu \), call the Lame parameters, and the elastic wave equation simplifies to

\[
\rho \frac{\partial^2 \mathbf{u}_i}{\partial t^2} = \sum_{j} \frac{\partial}{\partial x_j} \left( \mu \frac{\partial \mathbf{u}_i}{\partial x_j} + (\lambda + \mu) \frac{\partial \mathbf{u}_j}{\partial x_i} \right) + \mathbf{F}_i.
\]

In vector form, the elastic wave equation becomes

\[
\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla((\lambda + \mu) \nabla \cdot \mathbf{u}) + \mathbf{F}.
\]

\(^3\)We are ignoring a small advection term that transports moment, for the material velocity is much smaller than the wave velocity.
12. Special solutions to the elastic wave equation

If we have time, we should try to find exact solutions for P-waves

\[ u(x, y, z, t) = (f(t - x/\alpha), 0, 0) \]

and S-waves

\[ u(x, y, z, t) = (0, f(t - x/\beta), 0). \]

We should find \( \alpha = \sqrt{(\lambda + 2\mu)/\rho} \) and \( \beta = \sqrt{\mu/\rho} \), when all the parameters are constant. Maybe we can get some nice answers for certain varying parameters, like with the one-way equations.

13. One-way wave propagators

A wave propagator, or wave field extrapolator, is any method used to estimate the derivative of the wave field in one direction (say in depth), and thus use it to extrapolate the wave field at a deeper depth. That is, we want to make an estimate of the form

\[ u(z + \Delta z) \approx u(z) + \Delta z \frac{du}{dz}. \]

Given a two-way wave equation, it is often possible to approximately factor the differential operators into two separate ones, one representing waves going upwards, and another for waves going downwards.

Note there is a wide variety of propagators to choose from: implicit and explicit methods in the space domain, phase shift method in the wavenumber domain. In mixed domain, there is phase-shift plus interpolation, split-step and its extensions, Fourier finite-difference, generalized-screen, and others. Here we will look at some explicit and phase shift methods.

With the acoustic wave equation (in 2D for simplicity)

\[ \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} - \nabla^2 p = 0, \]

we separate out the \( z \)-derivative to obtain

\[ \frac{\partial^2 p}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} - \frac{\partial^2 p}{\partial x^2}. \]

If we can make sense of the square root of a differential operator, we could write

\[ \frac{\partial p}{\partial z} = \left( \sqrt{\frac{1}{c^2} \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2}} \right) p. \]

Claerbout suggests modelling a nearly-plane wave solution as \( p(x, z, t) = Q(x, z)e^{-i\omega(t - z/c)} \)
and inserting into the wave equation to obtain a parabolic equation

\[ \frac{\partial Q}{\partial z} + \frac{c}{2i\omega} \frac{\partial^2 Q}{\partial x^2} = 0, \]
which he calls the $15^\circ$ equation. A higher order equation, with more accurate interpolation (called the $45^\circ$ equation) is given by

$$\frac{\partial^2 Q}{\partial x^2} + \frac{2i\omega}{c} \frac{\partial Q}{\partial z} - \frac{c}{2i\omega} \frac{\partial^3 Q}{\partial z \partial x^2} = 0.$$  

This equation is first order in $z$, and a step-wise numerical extrapolation can be made in the $z$ direction. While the mathematical manipulations assumes the velocity $c$ is constant, Claerbout points out that the method works quite well even when $c(x,z)$ is not constant.

An alternative method of producing an extrapolator is to Fourier transform in the $x,t$ variables to obtain a frequency domain representation of the wave field, $P(k_x,z,\omega)$ which transforms equation (48) into

$$\frac{\partial^2 P}{\partial z^2} = \left(k_x^2 - \frac{\omega^2}{c^2}\right) P.$$  

When the velocity is constant, we take square roots to obtain a DE in $z$,

$$\frac{\partial P}{\partial z} = \left(-\sqrt{k_x^2 - \frac{\omega^2}{c^2}}\right) P,$$

where we have explicitly chosen a negative square root so that we don’t obtain exponentially growing solution. We now solve this ODE in $z$ explicitly, as

$$P(k_x,z + \Delta z,\omega) = \exp(-\Delta z \sqrt{k_x^2 - \frac{\omega^2}{c^2}}) P(k_x,z,\omega).$$

When the velocity $c = c(x)$ is laterally varying, this square root operator is not properly defined. However, two approaches give reasonable approximations. The first Fourier transforms back into the $x$ variable, yielding the wave field in the frequency domain,

$$\hat{p}(x, z + \Delta z, \omega) = \int \exp(-\Delta z \sqrt{k_x^2 - \frac{\omega^2}{c^2}}) P(k_x,z,\omega) e^{ik_x x} dk_x,$$

where $\hat{p}(x,z,\omega)$ is just the Fourier transform of $p(x,z,t)$ over the time variable. This approach to extrapolation is the phase shift plus interpolation method (PSPI).

The second method applies the exponential factor before taking the Fourier transform in $x$, to obtain

$$P(k_x,z + \Delta z,\omega) = \frac{1}{2\pi} \int \exp(-\Delta z \sqrt{k_x^2 - \frac{\omega^2}{c(x)^2}}) \hat{p}(x,z,\omega) e^{-ik_x x} dx.$$  

This is the non stationary phase shift method (NSPS).

The function $\alpha(x,k_x,\omega) = \exp(-\Delta z \sqrt{k_x^2 - \frac{\omega^2}{c(x)^2}})$ is called the symbol for these Fourier integral operators. It is worth noting that these two methods are adjoints of each other.
14. Numerical solutions for propagators

The differential equations in Eqns (50) and (51) are first order in \( z \), so a straightforward finite difference code will solve this quickly enough.

For something a bit different, let's look at implementing the wave field propagator given in Eqn (54). The steps are simple enough, and fast because they require only a couple of Fourier transforms and a element-by-element multiplication of two arrays.

Beginning with the initial wave field at \( z = 0 \), say \( p(x, 0, t) \), take the discrete Fourier transform in \( x \) and \( t \) to obtain the frequency representation \( P(k_x, \omega) \). Multiply this point-wise by the array of values \( \exp(-\Delta z \sqrt{k_x^2 - \omega^2/c^2}) \). This gives the wave field at \( z = \Delta z \), in the frequency domain. Invert the Fourier transform, and you have your extrapolated wave field.

Some code, in the Julia language, to implement this is given below. In the code, \( k \) is a vector of the wavenumber values, and \( \omega \) is the vector of the frequencies in radians/sec. \( p_0 \) is the array representing the initial wave field in variables \( x \) and \( t \), while \( P \) is its Fourier transform. Note that Julia is very swift about converting vectors into arrays when needed. The introduction of the complex number \( im \) is to remind Julia to compute complex square roots when needed.

```julia
# some code in Julia.
k = (k.^2)' .- (w/c).^2
alpha = exp(-150*sqrt(kk+0*im))
P = fft(p0,1:2)
P = alpha.*P
p=real(ifft(P,1:2))
imshow(p)
# end of code
```

The results of extrapolating an initial impulse at time \( t = 0 \), position \( x = 500m \) to two different depths are shown in Figure 5.

With this basic code in place, it is not too hard to implement NSPS or PSPI. However, the integral involved is no longer a simple FFT, so these extrapolators are numerically more costly. With apologies to Margrave and Ferguson, Fig 6 show a result for a lateral velocity step.
Here is the Julia code used to create the figures in these notes.
First, code to produce exact solutions to the 1D one-way wave equation.

# This code produces a right moving wave, for the PDE \( \frac{du}{dt} = c(x) \frac{du}{dx} \).
# Gives an exact solution, using slowness curves.
# I also produce a second figure for PDE \( \frac{du}{dt} = \frac{d}{dx} (c(x)u) \), which will show amplitude variations

using PyPlot

hold(0)

xleft=-2 # I needed just a little more space on the left
xright=6 # here is the right side
kink = 2 # we want the minimum velocity at kink

# this is the antiderivative of the velocity \( a(x) \) in the wave equation
function Slowness(x)
    return (x + 4*(atan(1*(x-kink))-atan(0-kink))) # normalized so Slowness(0) = 0
end

function c(x) # velocity = c(x). the reciprocal of the derivative of slowness
    return 1 ./((1 + 4 ./((x-kink).^2)))
end

x = linspace(xleft,xright,500)
t = Slowness(x)
Figure 6. Extrapolation thru a velocity step in x. NSPS and PSPI.

function mywave(x) max(0,1-(x-1).^2) end

plot (x,mywave(Slowness(x)))
hold(0)
for j=0:5
    plot(x,t+j,"k")
    hold(1)
end

plot(x,4.5*mywave(0-Slowness(x)),"r")
plot(x,5+4.5*mywave(5-Slowness(x)),"r")
plot(x,10+4.5*mywave(10-Slowness(x)),"r")
plot(x,15+4.5*mywave(15-Slowness(x)),"r")

    title("Rate of cars")
    xlabel("Position")
    ylabel("Time")

    plot(x,3*mywave(0-Slowness(x))./c(x),"r")
    plot(x,5*3*mywave(5-Slowness(x))./c(x),"r")
    plot(x,10+3*mywave(10-Slowness(x))./c(x),"r")
    plot(x,15+3*mywave(15-Slowness(x))./c(x),"r")

    title("Density")
    xlabel("Position")
    ylabel("Time")

    Second, finite difference code for numerical solutions to the 1D one-way wave equation.

    using PyPlot
    hold(0)

    xleft=-2 # I needed just a little more space on the left
    xright=6 # here is the right side
    kink = 2 # we want the minimum velocity at kink

    # this is the antiderivative of the velocity a(x) in the wave equation
    function Slowness(x)
      return (x + 4*(atan(1*(x-kink))-atan(0-kink))) # normalized so Slowness(0) = 0
    end

    function c(x) # velocity = c(x). the reciprocal of the derivative of slowness
      return 1 ./((1 + (4 ./((1 + (x-kink).^2))
    end

    function mywave(x) max(0,1-(x-1).^2) end

    # Set up the variables. We have x and t and vector u = u(x), which we will change over time
    dx = .001
    dt = .001

    x = xleft:dx:xright
    cx=c(x); # precompute the velocities

    u = mywave(0-Slowness(x))
    hold(1)
    plot(x,4.5*u,"r")
    for k=1:3
      for j=1:5000
\begin{verbatim}
  u = u - (dt/dx)*cx.*[0,diff(u)]  # placing 0 on the other side of diff causes instability
  plot(x,5*k+4.5*u,"r")
end

end

plot(x,5*k+4.5*u,"r")
end

title("Rate of cars (dt=0.01 and dt=0.001)")
xlabel("Position")
ylabel("Time")

# I can do the Lax-Wendroff scheme here as well

dx = .01
dt = .005

x = xleft:dx:xright
cx=c(x);  # precompute the velocities
R2 = (dt/dx)*cx
Rsq2 = 2*(R2).^2

u = mywave(0-Slowness(x))
hold(1)
plot(x,4.5*u,"r")
for k=1:3
  for j=1:1000
    u = u - R2.*[0,diff(u)] + Rsq2.*[0,diff(diff(u)),0]
  end
  plot(x,5*k+4.5*u,"r")
end

end

end

title("Rate of cars (Lax-Wendroff scheme, dt = 0.005)")
xlabel("Position")
ylabel("Time")

Finally, code to do the wave field extrapolation.

using PyPlot

\begin{verbatim}
c = 1000  # velocity in m/s
L = 1000  # maximum length for the x variable, in meters
T = 2     # maximum time length
dz = 150  # depth step (large), in meters

Nx = 1024 # grid points in x
Nt = 2048 # grid points in t

x = linspace(0,L,Nx)  # grid points for x
t = linspace(0,T,Nt)  # grid points for t
k = linspace(0,2*pi*Nx/L,Nx)  # wave number space
\end{verbatim}
\end{verbatim}
k[Nx:(-1):(Nx/2 +1)] = -k[2:(Nx/2 +1)] \ # put in negative wavenumbers explicitly
w = linspace(0,2*pi*Nt/T,Nt)
w[Nt:(-1):(Nt/2 +1)] = -w[2:(Nt/2 +1)] \ # put in negative frequencies explicitly

f0 = 40 \ # frequency for damped sine wave, in Hz
d0 = .01 \ # damping length for damped sine wave, in sec
s = sin(2*pi*t).*exp(-t/d0) \ # time waveform

w0 = 10 \ # half width for the pulse, in x direction (in meters)
10 = L/2 \ # the point where the pulse happens
r = max(0,1-((x-l0)/w0).^2)

p0 = broadcast(*,r',s) \ # the initial wave field, in time and space

kk = (k.^2)' .- (w/c).^2
alp = exp(-150*sqrt(kk+0*im))
P = fft(p0,1:2)
P = alp.*P
p=real(ifft(P,1:2))
imshow(p[1:Nt/2,:],extent=[0,1,1,0])
title("Depth step of 150m")
xlabel("Position (km")
ylabel("Time (sec")

References